BASE DE DATOS
Clase Jueves 30 de Mayo de 2025

Nombre: *Fabricio Josue Ruiz Aguilar*

APE: Álgebra Relacional y SQL
Modelo Relacional

```
CREATE DATABASE academia;
USE academia;

CREATE TABLE student(
id INT AUTO_INCREMENT PRIMARY KEY,
first_name VARCHAR(128) NOT NULL,
last_name VARCHAR(128) NOT NULL,
email VARCHAR(128) NOT NULL,
bith_date DATE NOT NULL,
start_date DATE NOT NULL
);

CREATE TABLE lecturer(
id INT AUTO_INCREMENT PRIMARY KEY,
first_name VARCHAR(128) NOT NULL,
last_name VARCHAR(128) NOT NULL,
degree VARCHAR(32) NOT NULL,
email VARCHAR(128) NOT NULL
);

CREATE TABLE course(
id INT AUTO_INCREMENT PRIMARY KEY,
title VARCHAR(128) NOT NULL,
learning_path VARCHAR(128) NOT NULL,
short_description VARCHAR(1200) NOT NULL,
lecture_hours INT NOT NULL,
tutorial_hours INT NOT NULL,
ects_points INT NOT NULL,
has_exam BOOLEAN NOT NULL,
has_project BOOLEAN NOT NULL
);

CREATE TABLE academic_semester(
id INT AUTO_INCREMENT PRIMARY KEY,
calendar_year INT NOT NULL,
term VARCHAR(128) NOT NULL,
start_date DATE NOT NULL,
end_date DATE NOT NULL
);

CREATE TABLE course_edition(
id INT AUTO_INCREMENT PRIMARY KEY,
course_id INT,
academic_semester_id INT,
lecturer_id INT,
FOREIGN KEY (course_id) REFERENCES course(id),
FOREIGN KEY (academic_semester_id) REFERENCES academic_semester(id),
FOREIGN KEY (lecturer_id) REFERENCES lecturer(id)
);
```

```
CREATE TABLE course_enrollment(
id INT AUTO_INCREMENT PRIMARY KEY,
midterm_grade DECIMAL(5,2) NOT NULL,
final_grade DECIMAL(5,2) NOT NULL,
course_letler_grade VARCHAR(3) NOT NULL,
passed BOOLEAN NOT NULL,
course_edition_id INT,
student_id INT,
FOREIGN KEY (course_edition_id) REFERENCES course_edition(id),
FOREIGN KEY (student_id) REFERENCES student(id)
);
```

Insertar Datos

student

```
INSERT INTO student (first_name, last_name, email, bith_date, start_date) VALUES
('Ana', 'Gómez', 'ana.gomez@email.com', '2002-03-15', '2021-09-01'),
('Luis', 'Martínez', 'luis.martinez@email.com', '2001-07-22', '2020-09-01'),
('María', 'Fernández', 'maria.fernandez@email.com', '2003-01-30', '2022-09-01'),
('Carlos', 'Ruiz', 'carlos.ruiz@email.com', '2000-12-10', '2019-09-01'),
('Elena', 'Torres', 'elena.torres@email.com', '2002-06-25', '2021-09-01'),
('Pedro', 'Díaz', 'pedro.diaz@email.com', '2001-09-05', '2020-09-01');
```

lecturer

```
INSERT INTO lecturer (first_name, last_name, degree, email) VALUES
('Juan', 'Lopez', 'PhD', 'juan.lopez@email.com'),
('Sofía', 'Morales', 'PhD', 'sofia.morales@email.com'),
('Miguel', 'Santos', 'MSc', 'miguel.santos@email.com'),
('Lucía', 'Reyes', 'PhD', 'lucia.reyes@email.com'),
('Andrés', 'Vega', 'MSc', 'andres.vega@email.com'),
('Paula', 'Cruz', 'PhD', 'paula.cruz@email.com');
```

course

```
INSERT INTO course (title, learning_path, short_description, lecture_hours, tutorial_hours,
ects_points, has_exam, has_project) VALUES
('Introducción a la Programación', 'Informática', 'Curso básico de programación en Python.',
30, 15, 6, TRUE, TRUE),
('Bases de Datos', 'Informática', 'Modelado, SQL y administración de bases de datos.', 40,
20, 7, TRUE, TRUE),
('Cálculo I', 'Ingeniería', 'Fundamentos de cálculo diferencial.', 45, 15, 6, TRUE, FALSE),
('Física General', 'Ingeniería', 'Mecánica y termodinámica.', 50, 10, 6, TRUE, FALSE),
('Diseño Web', 'Informática', 'HTML, CSS, JavaScript y herramientas modernas.', 25, 20, 5,
FALSE, TRUE),
('Ética Profesional', 'General', 'Estudio de la ética aplicada a la vida laboral.', 20, 10, 3,
FALSE, FALSE);
```

academic_semester

```
INSERT INTO academic_semester (calendar_year, term, start_date, end_date) VALUES
(2023, 'Primavera', '2023-02-01', '2023-06-30'),
(2023, 'Otoño', '2023-08-15', '2023-12-20'),
(2024, 'Primavera', '2024-02-01', '2024-06-30'),
(2024, 'Otoño', '2024-08-15', '2024-12-20'),
(2025, 'Primavera', '2025-02-01', '2025-06-30'),
(2025, 'Otoño', '2025-08-15', '2025-12-20');
```

course_edition

```
INSERT INTO course_edition (course_id, academic_semester_id, lecturer_id) VALUES
(1, 1, 1),
(2, 1, 2),
(3, 2, 3),
(4, 2, 4),
(5, 3, 5),
(6, 3, 6);
```

course_enrollment

```
INSERT INTO course_enrollment (midterm_grade, final_grade, course_letler_grade,
passed, course_edition_id, student_id) VALUES
(85.00, 90.00, 'A', TRUE, 1, 1),
(78.00, 82.00, 'B', TRUE, 1, 2),
(50.00, 65.00, 'C', TRUE, 2, 3),
(40.00, 50.00, 'D', FALSE, 2, 4),
(90.00, 95.00, 'A', TRUE, 3, 5),
(70.00, 75.00, 'B', TRUE, 4, 6);
```

Clases

Respaldo:

```
mysqldump -u michu -p academia > respaldo_academia.sql
mysql -u michu -p cargarp < ~/respaldo_academia.sql
```

Inner Join

```
SELECT dni, nombres, apellidos, nombre, edificio FROM empleado INNER JOIN
departamento ON departamento.id_departamento = empleado.id_departamento;
```

1. ***Mostrar los IDs y títulos de todos los cursos que tuvieron lugar durante cualquier trimestre de primavera.***

```
SELECT id, title FROM course WHERE id IN (SELECT course_id FROM course_edition ce
JOIN academic_semester acs ON ce.academic_semester_id = acs.id WHERE term =
'Primavera');
```

```
SELECT ce.course_id, c.title FROM course_edition ce JOIN academic_semester acs ON
ce.academic_semester_id = acs.id JOIN course c ON ce.course_id = c.id WHERE acs.term
= 'Primavera';
```

2. ***Seleccionar los IDs y nombres de los estudiantes que aprobaron al menos un curso.***

```
SELECT student.id, student.first_name, student.last_name FROM student JOIN
course_enrollment ce ON student.id = ce.student_id WHERE ce.passed = TRUE;
```

3. ***Encuentre el/los profesor/es con el menor número de cursos impartidos. Muestre el nombre y apellidos del profesor y el número de cursos que imparte***

```
SELECT lecturer.first_name, lecturer.last_name, COUNT(*) AS total FROM course_edition,
lecturer WHERE course_edition.lecturer_id = lecturer.id GROUP BY lecturer.id ORDER BY
total LIMIT 1;
```